Listing of claims:

1 (currently amended): A computer-implemented method for a client to interact with a server, the computer-implemented method comprising:

creating a cached object from an original object, the original object being managed by the server; wherein the client interacts with a plurality of cached objects that are created on the client from objects managed by the server;

creating a bond manager on the server, the bond manager including:

a filter component configured to scan for incoming and outgoing events related to shared files from a file system manager of the server and forward the events to a service component; and

the service component configured to receive events from the filter component and establish notification bonds with the client, the service component maintaining a server bond table, wherein the server bond table includes a server identifier, a server aggregate bond number, and a notification log offset identifying a location within the notification log;

establishing a notification bond associated with a particular object with the server, the notification bond enabling the client to obtain a notification from the bond manager on the server in response to an object related event associated with the original object; wherein the notification bond remains persistent through a reboot of the client and server and the object related event is associated with an edit of the original object, the notification bond established after determining the notification bond to be missing by comparing a client aggregate bond number with the server aggregate bond number; and wherein each object is associated with a different ~~includes a~~ notification bond; and

updating ~~each of~~ the cached object ~~objects~~ with a change to the original ~~objects~~ object after [[a]] the change is made to the original object.

2 (original): The computer-implemented method of Claim 1, wherein the object related event includes when the original object has been modified.

3 (original): The computer-implemented method of Claim 1, wherein establishing the notification bond is performed in response to creating the cached object.

4 (original): The computer-implemented method of Claim 1, further comprising:
obtaining a notification from the server; and
updating the cached object using the notification.

5 (original): The computer-implemented method of Claim 1, wherein obtaining the notification includes retrieving a notification log containing the notification.

6 (original): The computer-implemented method of Claim 1, further comprising:
reconnecting with the server after a disconnected period of time;
requesting a notification log containing a notification; and
synchronizing the cache object with the original object using the notification.

7 (original): The computer-implemented method of Claim 1, further comprising maintaining states associated with the notification bond.

8 (original): The computer-implemented method of Claim 7, wherein the states are maintained in a persistent medium.

9 (original): The computer-implemented method of Claim 8, wherein the states include a bond number that uniquely identifies the notification bond.

10 (original): The computer-implemented method of Claim 8, wherein the states include an aggregate bond number that is unique to the client.

11 (original): The computer-implemented method of Claim 8, further comprising reestablishing the states on the client after a restart.

12 (original): The computer-implemented method of Claim 11, further comprising synchronizing the states on the client and corresponding states on the server.

13 (currently amended): A computer-implemented method for a server to interact with a client, the computer-implemented method comprising:

creating a bond manager on the server, the bond manager including:

a filter component configured to scan for incoming and outgoing events related to shared files from a file system manager of the server and forward the events to a service component; and

the service component configured to receive events from the filter component and establish notification bonds with the client, the service component maintaining a server bond table, wherein the server bond table includes a server identifier, a server aggregate bond number, and a notification log offset identifying a location within the notification log;

establishing a notification bond associated with a particular object with the client, the notification bond enabling the client to obtain a notification from the server in response to an object related event associated with an object; wherein the notification bond associated with a particular object remains persistent through a reboot and the object related event is associated with an edit of the original object, the notification bond established after determining the notification bond to be missing by comparing a client aggregate bond number with the server aggregate bond number; and

enabling the client to cache the object.

14 (original): The computer-implemented method of Claim 13, wherein the object related event includes when the object has been modified.

15 (original): The computer-implemented method of Claim 13, wherein establishing the notification bond is performed in response to a request from the client to cache the object.

16 (original): The computer-implemented method of Claim 13, further comprising:
determining an object related event that was not caused by the client;
creating a notification in accordance with the notification bond; and
providing the notification to the client.

17 (original): The computer-implemented method of Claim 13, further comprising:
determining an object related event that was not caused by the client;
creating a notification in accordance with the notification bond; and
recording the notification in a notification log.

18 (original): The computer-implemented method of Claim 17, further comprising:
establishing a connection with the client; and
sending the notification log to the client.

19 (currently amended): A distributed file system for sharing objects, comprising:
a server that includes a processor and storage medium encoded with instructions to manage original objects, the server including a bond manager configured to issue notification bonds to clients, the bond manager comprising:

a filter component configured to scan for incoming and outgoing events related to shared files from a file system manager of the server and forward the events to a service component; and

the service component configured to receive events from the filter component and establish notification bonds with the client, the service component maintaining a server bond table, wherein the server bond table includes a server identifier, a server aggregate bond number, and a notification log offset identifying a location within the notification log;

and each notification bond being associated with a particular original object and enabling a client to obtain a notification from the server in response to an object related event associated with [[an]] the particular original object in which the notification bond is associated; wherein the notification bond associated with the particular ~~objects~~ object remains persistent through a reboot and the object related event is associated with an edit of the original object.

20 (original): The distributed file system of Claim 19, wherein the bond manager is configured to provide notifications to the clients in accordance with the notification bonds.

21 (previously presented): The distributed file system of Claim 19, wherein the storage medium of the server further comprises instructions that describe a file system manager configured to manage the original objects and wherein the bond manager comprises a filter component configured to determine object related events by monitoring communication traffic associated with the file system manager.

22 (original): The distributed file system of Claim 19, wherein the bond manager is configured to maintain a bond table and wherein the bond table includes states that relate each notification bond with an original object and a client to whom the notification is to be provided.

23 (original): The distributed file system of Claim 19, wherein the bond manager is configured to maintain a notification log and wherein the notification log includes notifications for the client.

24 (original): The distributed file system of Claim 19, further comprising:

a client configured to create a cached object associated with an original object managed by the server, the client including a notification handler configured to maintain a notification bond associated with the original object in conjunction with the server.

25 (original): The distributed file system of Claim 19, wherein the notification handler is configured to obtain from the server a notification log associated with the notification bond and to update the cached object in accordance with the notification log.

26 (original): The distributed file system of Claim 25, wherein the notification log includes notifications associated with a plurality of notification bonds.

27 (original): The distributed file system of Claim 19, wherein the notification handler is configured to maintain a bond table and wherein the bond table includes states that relate each notification bond with a cached object and a server that manages an original object corresponding to the cached object.

28 (currently amended): A computer-readable medium encoded with a data structure, comprising:

a first indexing data field containing object identifiers, each object identifier uniquely identifying an object that is managed by a server; and

a second data field containing entries, each entry being indexed to an object identifier in the first indexing data field and containing states associated with a notification bond associated with a particular object between the server and a client that caches the object identified by the object identifier; wherein the first indexing data field and the second data field are created by the server and wherein the server accesses the first indexing data field and the second data field to determine what objects on the client require notification in response to an object related edit event and wherein the notification bond associated with the particular object remains persistent through a reboot, the notification bond established after determining the notification bond to be missing by comparing a client aggregate bond number with a server aggregate bond number.

29 (original): The computer-readable medium of Claim 28, wherein each object identifier in the first indexing data field includes a file path associated with an object.

30 (original): The computer-readable medium of Claim 29, wherein each object identifier includes a hash of the file path.

31 (original): The computer-readable medium of Claim 28, wherein each entry of the second data field includes a bond number that uniquely identifies a notification bond.

32 (original): The computer-readable medium of Claim 28, wherein each entry of the second data field includes a client identifier that identifies a client associated with a notification bond.

33 (original): The computer-readable medium of Claim 28, wherein each entry of the second data field includes a type identifier that identifies a type associated with a notification bond.

34 (currently amended): A computer-readable medium encoded with a data structure, comprising:

a first indexing data field containing server identifiers, each server identifier uniquely identifying a server that manages an original object, the original object being cached by a client; and

a second data field containing entries, each entry being indexed to a server identifier in the first indexing data field and containing states associated with a notification bond associated with a particular object between the client and a server identified by the server identifier, the notification bond being associated with a cached object created by the client from an original object; wherein the first indexing data field and the second data field are accessed by a computing device and wherein the computing device uses the first indexing data field and the second data field in updating objects in response to an object related edit event and wherein the notification bond associated with the particular object remains persistent through a reboot, the notification bond established after determining the notification bond to be missing by comparing a client aggregate bond number with a server aggregate bond number.

35 (original): The computer-readable medium of Claim 34, wherein each entry of the second data field includes a bond number that uniquely identifies a notification bond.

36 (original): The computer-readable medium of Claim 34, wherein each entry of the second data field includes an original object identifier that identifies an original object.

37 (original): The computer-readable medium of Claim 34, wherein each entry of the second data field includes a cached object identifier that identifies a cached object associated with an original object.

38 (original): The computer-readable medium of Claim 34, further comprising a third data field that includes an aggregated bond number.

39 (currently amended): The computer-readable medium of Claim 34, further comprising a third data field that includes a notification log offset, wherein the notification log offset includes a location within a notification log.

40 (currently amended): A distributed file system for sharing objects, comprising:

means for a client to cache an original object managed by a server; wherein the client includes means for interacting with a plurality of cached objects that are created on the client from objects managed by the server and

means for establishing a notification bond associated with a particular original object with the server and the client, the notification bond enabling the client to obtain a notification from the server in response to an object related event associated with the particular original object in which the notification bond is associated; wherein the notification bond remains persistent through a reboot, the notification bond established after determining the notification bond to be missing by comparing a client aggregate bond number with a server aggregate bond number and the object related event is associated with an edit of the original object; and

updating each of the cached objects with the original objects after a change is made to the original object.

41 (original): The distributed file system of Claim 40, further comprising:
means for obtaining a notification from the server; and
means for updating the cached object using the notification.

42 (original): The distributed file system of Claim 40, further comprising:
means for reconnecting with the server after a disconnected period of time;
means for requesting a notification log containing a notification; and
means for synchronizing the cache object using the notification.

43 (original): The distributed file system of Claim 40, further comprising:
means for determining an object related event;
means for creating a notification in accordance with the notification bond; and
means for providing the notification to the client.

44 (original): The distributed file system of Claim 40, further comprising:
means for determining an object related event;
means for creating a notification in accordance with the notification bond; and
means for recording the notification in a notification log.

45 (original): The distributed file system of Claim 40, further comprising:
means for establishing a connection with the client; and
means for sending the client the notification log.

46 (original): The distributed file system of Claim 40, further comprising:
means for the client to drop the notification bond.

47 (original): The distributed file system of Claim 40, further comprising:
means for the server to drop the notification bond.

48 (original): The distributed file system of Claim 40, further comprising:
means for the server to drop all notification bonds associated with the client.

49 (original): The distributed file system of Claim 48, further comprising:
means for the server to reset clear states associated with the dropped notification bonds.

50 (currently amended): A computer-implemented method for maintaining cached objects that correspond to original objects managed by a server, the computer-implemented method comprising:

creating cached objects from original objects;

establishing a plurality of notification bonds, each associated with a particular one of the cached objects, between the server and the computer, the notification bonds enabling the client to obtain a notification from the server in response to an object related event associated with the original object in which the notification bond is associated; wherein the plurality of notification bonds between the server and the computer that are associated with one of the ~~cashed~~ cached objects each remain[[s]] persistent through a reboot of the client and server and the object related event is associated with an edit of the original object; and wherein each object includes is associated with a different one of the plurality of [[a]] notification bonds;

reestablishing a communication link between the client and the server after a period of time without a communication link;

determining at least one notification bond is missing by comparing a client aggregate bond number with a server aggregate bond number;

re-establishing the at least one notification bond;

obtaining notifications from the server about changes made to at least one of the original objects during the period of time; and

synchronizing a cache object corresponding to the at least one original object using the notifications, without synchronizing all of the cached objects.

51 (currently amended): A computer-implemented method for synchronizing cached objects maintained by a client with the corresponding original objects maintained by a server, the computer-implemented method comprising:

creating cached objects from original objects;

establishing a notification bond associated with one of the original objects with the server for each of the cached objects, the notification bond enabling the client to obtain a notification from the server in response to an edit associated with the original object;

determining the notification bond is missing by comparing a client aggregate bond number with a server aggregate bond number, wherein the notification bond associated with the original objects remains persistent through a reboot of the client and server and the edit is associated with a modification of the original object; wherein each object includes a notification bond; and wherein the edit to the object is one of an edit to a word processing document, an edit to a spreadsheet document; or an edit to an image file;

persistently maintaining, by the server, a server identifier, a server aggregate bond number, a notification log offset identifying a location within the notification log, and server bond states related to the original objects,; the server bond states corresponding to the notification bonds associated with the original objects, each notification bond enabling the client to obtain a notification from the server when at least one of the original object has been modified such that a copy of the original object may be synchronized and maintained on the client after the object has been edited by a user associated with the server; and

persistently maintaining, by the client, client bond states corresponding to the server bond states.

52 (original): The computer-implemented method of claim 51, further comprising reestablishing the server bond states after a server reboot or restart.

53 (original): The computer-implemented method of claim 52, further comprising recovering notifications associated with the notification bonds after a server reboot or restart.

54 (original): The computer-implemented method of claim 51, further comprising reestablishing the client bond states after a client reboot or restart.

55 (original): The computer-implemented method of claim 54, further comprising recovering notifications associated with the notification bonds after a server reboot or restart.

56 (original): The computer-implemented method of claim 51, further comprising
determining, by the server, to drop a notification bond;
performing, by the server, an operation to drop the notification bond; and
providing, by the server, a notification to the client for dropping the notification bond.

57 (original): The computer-implemented method of claim 56, wherein performing the operation commits the server to dropping the notification bond.

58 (original): The computer-implemented method of claim 57, wherein performing the operation is completed before providing the notification to the client.

59 (original): The computer-implemented method of claim 51, further comprising
determining, by the server, to drop all notification bonds associated with the client;
performing, by the server, an operation to drop the notification bonds; and
providing, by the server, a notification to the client for dropping the notification bonds.

60 (original): The computer-implemented method of claim 59, wherein performing the operation commits the server to dropping the notification bonds.

61 (original): The computer-implemented method of claim 60, wherein performing the operation is completed before providing the notification to the client.

62 (original): The computer-implemented method of claim 51, further comprising
determining, by the client, to drop a notification bond;
performing, by the client, an operation to drop the notification bond; and
requesting the server to drop the notification bond.

63 (original): The computer-implemented method of claim 62, wherein performing the operation commits the client to dropping the notification bond.

64 (original): The computer-implemented method of claim 63, wherein performing the operation is completed before requesting the server to drop the notification bond.